Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/EP04/014187

International filing date: 13 December 2004 (13.12.2004)

Document type: Certified copy of priority document

Document details: Country/Office: EP
Number: 04090010.2
Filing date: 14 January 2004 (14.01.2004)

Date of receipt at the International Bureau: 17 February 2005 (17.02.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)

World Intellectual Property Organization (WIPO) - Geneva, Switzerland
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse

14187
13.12.04

Europäisches Patentamt	European Patent Office	Office européen des brevets

Bescheinigung | Certificate | Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.	The attached documents are exact copies of the European patent application described on the following page, as originally filed.	Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. | Patent application No. | Demande de brevet n°

04090010.2

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

R C van Dijk

Europäisches Patentamt | European Patent Office | Office européen des brevets

PCT/EP2004/014187

13.12.04

Anmeldung Nr:
Application no.: 04090010.2
Demande no:

Anmeldetag:
Date of filing: 14.01.04
Date de dépôt:

Anmelder/Applicant(s)/Demandeur(s):

DEUTSCHE THOMSON-BRANDT GMBH
Hermann-Schwer-Strasse 3
78048 Villingen-Schwenningen
ALLEMAGNE

Bezeichnung der Erfindung/Title of the invention/Titre de l'invention:
(Falls die Bezeichnung der Erfindung nicht angegeben ist, siehe Beschreibung.
If no title is shown please refer to the description.
Si aucun titre n'est indiqué se referer à la description.)

Method for generating an on-screen menu

In Anspruch genommene Priorität(en) / Priority(ies) claimed /Priorité(s) revendiquée(s)
Staat/Tag/Aktenzeichen/State/Date/File no./Pays/Date/Numéro de dépôt:

Internationale Patentklassifikation/International Patent Classification/
Classification internationale des brevets:

G06F3/033

Am Anmeldetag benannte Vertragstaaten/Contracting states designated at date of filing/Etats contractants désignées lors du dépôt:

AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LU MC NL
PT RO SE SI SK TR LI

PD040020-IPA (Kö-2004-01-14)

Method for generating an on-screen menu

EPO-BERLIN
14 -01- 2004

Field of the invention

This invention relates to a method for generating an on-screen menu.

Background

Audio-visual contents of data storage media, e.g. Digital Versatile Discs (DVD) for video applications, usually contain menu data for various applications, e.g. to enable a user to select specific content of the medium. The menu data are used for rendering the menu on a display screen. Often so-called multi-page menus are used, where each possible state of the menu is represented by a full-screen image that is overlaid as a separate layer to the video picture. The menu layer is usually transparent, except for the displayed menu items.

In state-of-the-art menus the menu items basically consist of a number of buttons. Each button is assigned an on-screen position by the content author and can be navigated and activated by the user, e.g. via a remote control. Each button is associated a state, which can either be the 'normal' (or 'unselected') state, the 'selected' state or the 'activated' state. Each button can provide a different visual representation in each state in order to give the user feedback.

However, these kinds of menus are rather static as there is no way to dynamically add or remove buttons from the screen, without re-rendering the whole screen. For content

authors such more sophisticated menu features would be desirable, for example for the design of sub-menus. In such a case, additional buttons dynamically appear and disappear on the screen through user interaction.

The invention provides means to generate such dynamic menus.

Summary of the Invention

The present invention is based on the assumption that an on-screen menu is rendered not using the multi-page method, but the different menu items are rendered separately.

According to the invention, each button is assigned an additional state, which can either be the 'enabled' or the 'disabled' state. As a general rule, this state defines the rendering behaviour of the button. Buttons that are in the 'enabled' state are displayed on the screen, while buttons that are in the 'disabled' state are not displayed.

The user can navigate only buttons that are in the 'enabled' state, and their well known 'normal', 'selected' or 'activated' state is only valid within the 'enabled' state. The user cannot navigate buttons that are in the 'disabled' state. Any attempt to do that is ignored by the menu decoder according to the invention.

Each button within the menu is assigned an on-screen area and a unique identifier. Usually the on-screen area will be rectangular.

In the current invention, buttons are organized in groups, and all buttons obey to certain rules, which are described in the following.

- The number of buttons belonging to one button group can be one or more. There are no empty button groups.
- A button cannot belong to more than one button group.
- The on-screen area of any button belonging to a first button group does not overlap with the on-screen area of any other button that is not belonging to the same button group.
- Each button within a button group must be in one of the two states: 'enabled' or 'disabled'.
- Each button is assigned an initial state, which is either the 'enabled' or the 'disabled' state.
- Not more than one button within a button group can be in the 'enabled' state at a time, i.e. rendered on the screen. The 'enabled' state does not imply user-visibility, e.g. if it is represented only through transparent pixels; in this case an enabled button is not visible to the user.
- For each button within a button group there is defined neighbourhood information for button navigation, saying e.g. which other button to select when the user presses the LEFT, RIGHT, UP or DOWN button. This neighbourhood information is only valid when the button is in the 'enabled' state. The user cannot navigate to disabled buttons.
- The on-screen areas of a first button of a first button group and a second button of the same group, i.e. their visible representations, may overlap. They will not be visible simultaneously since they belong

to the same button group, and only one of them can be in the 'enabled' state at a time.

Further, the invention discloses a new command. This command is used to dynamically switch between the 'enabled' and the 'disabled' state of a button. In state-of-the-art menus, activating a button already may encompass the execution of one or more commands. The proposed command is activated in the same way and is therefore compatible with the state-of-the-art framework. Other effects of activating a button are commonly that the button changes its appearance, colour etc.

- According to the invention, a command for enabling and disabling a button is defined. The information about which button to enable or disable is provided through the button identifier as a parameter of the command.
- For each button there can be defined button commands that are executed on activation of the button. The execution of button commands is only possible when the button is in the 'enabled' state. There may be buttons however that have no associated button command.
- The disabling of a button may clear the button visibility by substituting it with transparent pixels.
- The inventive button command does not change the 'enabled' or 'disabled' state of its own button. This means that if an enabled/visible button is activated, the corresponding button command that is executed upon activation may switch the 'enabled'/'disabled' state of other buttons, but it may not switch its own button to the 'disabled' state. There may however other commands be executed that e.g. disable the whole menu.

- Enabling one button of a group disables all other buttons within that group.

In other words, for each button group an area is defined where its buttons may be rendered. This area is in the following called a button group area. It is usually rectangular, but can in principle have other shapes. The invention covers embodiments according to the following possibilities:

1. No button position within a button group may overlap with any other button position of a button belonging to another button group. The button group area of different button groups may not overlap at all. This means that the screen can be viewed upon as a number of non-overlapping button group areas, wherein a button group area may comprise several partial areas, and in each button group area a button belonging to the respective button group may be rendered visible or not.
2. An area, e.g. rectangular, is defined for each button group, which area comprises all possible button positions of buttons belonging to that button group. The areas that belong to different button groups may not overlap. The button needs not necessarily fill the allowed area, i.e. the button needs not have the size and shape of the button group area, but it must be fully within the area corresponding to its group. Therefore, buttons belonging to different groups may not overlap. Further, it is easy to fully delete a first button belonging to a first button group when displaying a second button belonging to the same button group, because in this case only the button

group area belonging to the respective button group needs to be re-rendered, which is a contiguous area; it is not necessary to re-render other parts of the screen. Thus, no remains of the previously shown button are visible. All buttons within a button group occupy the same on-screen area. This is the preferred embodiment.

3. All buttons of a button group have identical button positions on the screen. This is the easiest case, because rendering a button that belongs to a certain button group necessarily deletes another button of the same button group that was previously visible on the same position.

An apparatus that utilizes the method is disclosed in claim 2. A data storage medium holding a respective data structure is disclosed in claim 3.

Further objects, features and advantages of the invention will become apparent from a consideration of the following description and the appended claims when taken in connection with the accompanying drawings.

Brief description of the drawings

Exemplary embodiments of the invention are described with reference to the accompanying drawings, which show in

Fig.1-Fig.10 embodiments of the invention.

Detailed description of the invention

The invention provides more sophisticated menu features demanded by content authors. In particular, the invention provides means to generate dynamic menus, wherein buttons can be dynamically removed or added to a menu.

A content author is able to define hierarchical menus and sub-menus being represented by a flat data structure. The graphic decoder needs not know or handle the menu hierarchies. It simply has to handle isolated button groups instead.

Fig.1 shows a menu page with only an "Audio language" button and a "Subtitle Language" button visible. The other buttons are invisible, as defined through a data-segment that describes the initial menu and that is contained in a bitstream on the medium. The data segment can be called interactive composition segment (ICS). In this case, if e.g. the "Audio Language" button is selected and the RIGHT button is pressed, the button remains selected because there is no button to the right defined. Pressing the UP or DOWN button would select the "Subtitle Language" button. Activating the "Audio Language" button would lead to Fig.2.

In Fig.2 the "Audio Language" button is activated, and the associated button commands enable the buttons to the right. Consequently, these buttons are rendered visible, allowing to select and activate one of them, and thus to select a language.

Fig.3 shows a situation where e.g. starting from Fig.1 the "Audio Language" button was selected, but not activated,

and the DOWN button was pressed so that the "Subtitle Language" button gets selected, and the "Subtitle Language" button was activated. The effect is that the four buttons to the right are rendered in the same positions as the four buttons to the right were rendered in Fig.2. They are different though, since they have different functions, namely allow selection of a subtitle language.

Fig.4 shows a menu where a graphical user hint appears, in the form of a small arrow, showing that sub-items exist. When the "Audio Language" button is activated, then a virtual or invisible button being defined in the neighbourhood information as neighbour to the right, is selected and activated, therefore the menu handler is executed which leads to that the four buttons on the right are rendered visible, without any of them being activated. The virtual button has the same data structure as the other buttons. This means that the RIGHT button must be pressed when the "Audio Language" button is selected, and then the audio options become visible. In the next step, one of the new buttons on the right can be selected, as shown in Fig.5. There are six different button groups, namely 'Audio Language', 'Subtitle Language', 'English', 'Japanese', 'Korean' and 'Dutch'. In the shown situation, e.g. 'Audio Language' is selected, and the buttons representing the options for audio are visible. They are rendered at the same positions where also the buttons for subtitle options would be rendered, if the "Subtitle Languages" were selected. Therefore the buttons belonging to the same button group are overwriting each other. Also the "Subtitle Languages" button and the "Audio Language" button form button groups, but with only one button each.

Fig.6 shows the first possibility for definition of a button group, as described above. A first button group comprises three buttons, with three separate, non-coherent areas. A second button group also comprises three buttons, with three separate, non-coherent areas. None of the areas overlaps with any other area. In another embodiment though it would be possible that areas of buttons that belong to the same group may overlap.

Fig.7 shows the preferred embodiment, where a single area is defined for every button group, and all buttons belonging to that button group are positioned within that area. Button group areas of different button groups may not overlap. The position of different buttons that belong to the same button group may differ, as long as they are within the specified area.

Fig.8 shows a special case of the preferred embodiment, where all buttons of the same button group have exactly the same position, thus overwriting each other.

The group structure provides information for the menu decoder, the information defining which on-screen area needs update. Because within a button group not more than one button is active at a time, the activation of another button within a group implies the deactivation of the first button of the same group. This is an advantage for authoring, since it makes it easier to author menus.

Especially in the case of prerecorded media, e.g. prerecorded Blu-ray discs, a verification process is performed on any title before it is released to check if

the data-structure meets the specification. The invention allows for easy verification while providing enhanced features to the content author when creating dynamic menus.

Implementation example

In the following, the syntax of a data segment which is contained in a bitstream and which describes the initial menu screen, as shown in Tab.1, is described as an exemplary implementation. It describes the case that the button group area is defined according to "possibility 2" above.

Tab.1 : Exemplary Syntax of data segment describing a menu

	Syntax	No. of bits	Mnemonics
1	interactive_composition_segment() {		
2	**segment_type**	8	bslbf
3	**segment_length**	16	uimsbf
4	**composition_number**	16	uimsbf
5	**composition_state**	2	bslbf
6	reserved	6	bslbf
7	**command_update_flag**	1	bslbf
8	reserved	7	bslbf
9	**composition_time_out_pts**	33	uimsbf
10	reserved	7	bslbf
11	**selection_time_out_pts**	33	uimsbf
12	reserved	7	bslbf
13	**UO_mask_table()**	64	bslbf
14	**animation_frame_rate_code**	8	uimsbf
15	**default_selected_button_number**	8	uimsbf
16	**default_activated_button_number**	8	uimsbf
17	while (processed_length < segment_length) {		
18	button_group() {		
19	**button_group_horizontal_position**	16	uimsbf
20	**button_group_vertical_position**	16	uimsbf
21	**button_group_horizontal_size**	16	uimsbf
22	**button_group_vertical_size**	16	uimsbf
23	**default_enabled_button_number**	8	uimsbf
24	**num_buttons**	8	uimsbf
25	for (i=0; i<num_buttons;i++) {		

26	**button_number**	8	uimsbf
27	**numerically_selectable_flag**	1	bslbf
28	reserved	7	bslbf
29	**auto_action_flag**	1	bslbf
30	reserved	7	bslbf
31	**button_horizontal_position**	16	uimsbf
32	**button_vertical_position**	16	uimsbf
33	neighbor_info() {		
34	**upper_button_number**	8	uimsbf
35	**lower_button_number**	8	uimsbf
36	**left_button_number**	8	uimsbf
37	**right_button_number**	8	uimsbf
38	}		
39	normal_state_info() {		
40	**start_object_id_normal**	16	bslbf
41	**end_object_id_normal**	16	bslbf
42	**repeat_normal_flag**	1	bslbf
43	reserved	7	bslbf
44	}		
45	selected_state_info() {		
46	**start_object_id_selected**	16	bslbf
47	**end_object_id_selected**	16	bslbf
48	**repeat_selected_flag**	1	bslbf
49	reserved	7	bslbf
50	}		
51	actioned_state_info() {		
52	**start_object_id_activated**	16	bslbf
53	**end_object_id_activated**	16	bslbf
54	}		
55	**num_of_button_commands**	8	uimsbf

56	for(cmd_id = 0;
	cmd_id < num_of_button_commands;
	cmd_id++) {		

57	**button_command[command_id]**	96	bslbf
58	}		
59	}		
60	}		
61	}		
62	}		

For each of a number of button groups, their respective position on the screen is defined in lines 19-22. The parameter in 1.23 defines the number of buttons in a group. A loop beginning in 1.25 covers all buttons of the group

For every button, several commands can be specified for each button. The number of commands per button is specified in the parameter num_of_button_commands in line 55.

The invention is usable for menu decoders and menu data structures, e.g. for DVDs, Blu-ray discs or other media.

EPO-BERLIN
14 -01- 2004

Claims

1. A method for generating an on-screen menu comprising menu buttons and other menu items, e.g. help text, characterized in that
   - one or more groups of menu buttons or menu items are defined, a group comprising one or more menu buttons or menu items and having associated a defined area on the screen;
   - a state is assigned to a button, the state being "enabled" or "disabled", wherein only an enabled button or item may be displayed, and wherein within a group not more than one button or item may be enabled.

2. Apparatus performing a method according to claim 1.

3. Data storage medium containing a data structure according to claim 1.

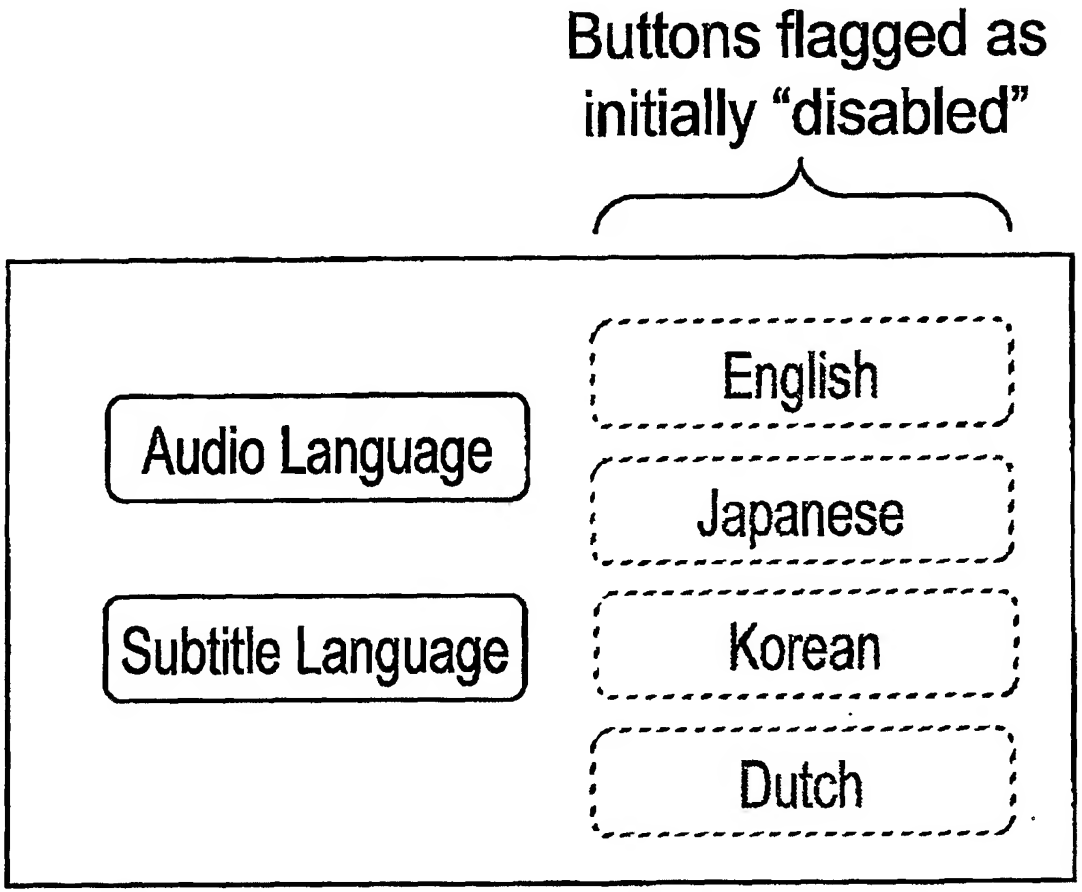
EPO-BERLIN
14 -01- 2004

Abstract

A method for generating an on-screen menu comprising menu buttons and other menu items, e.g. help text, is characterized in that one or more groups of menu buttons or menu items are defined, a group comprising one or more menu buttons or menu items and having associated a defined area on the screen; and a state is assigned to a button, the state being "enabled" or "disabled", wherein only an enabled button or item may be displayed, and wherein within a group not more than one button or item may be enabled.

Fig.1

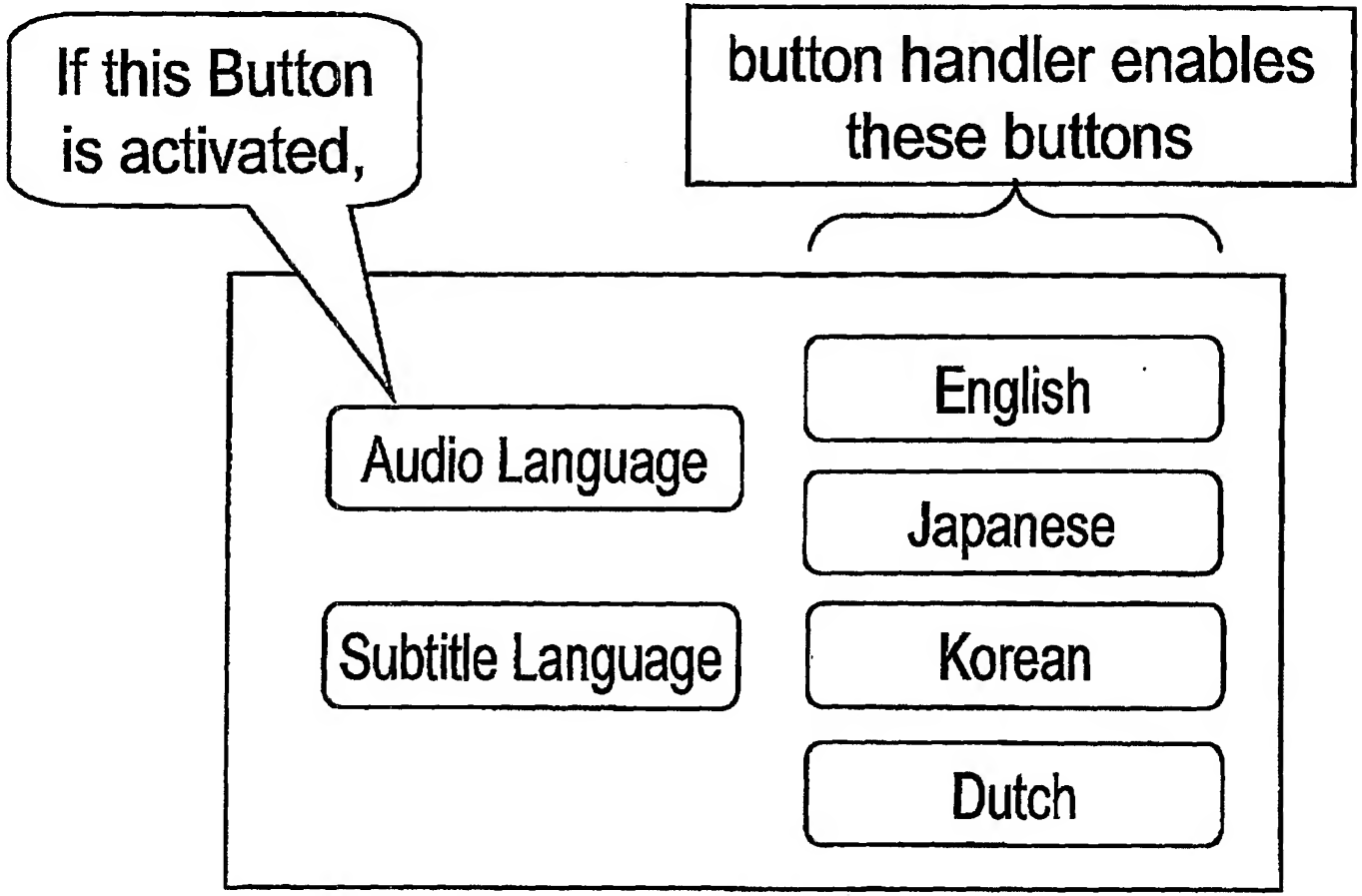
Explanation: Enable/Disable Button Command

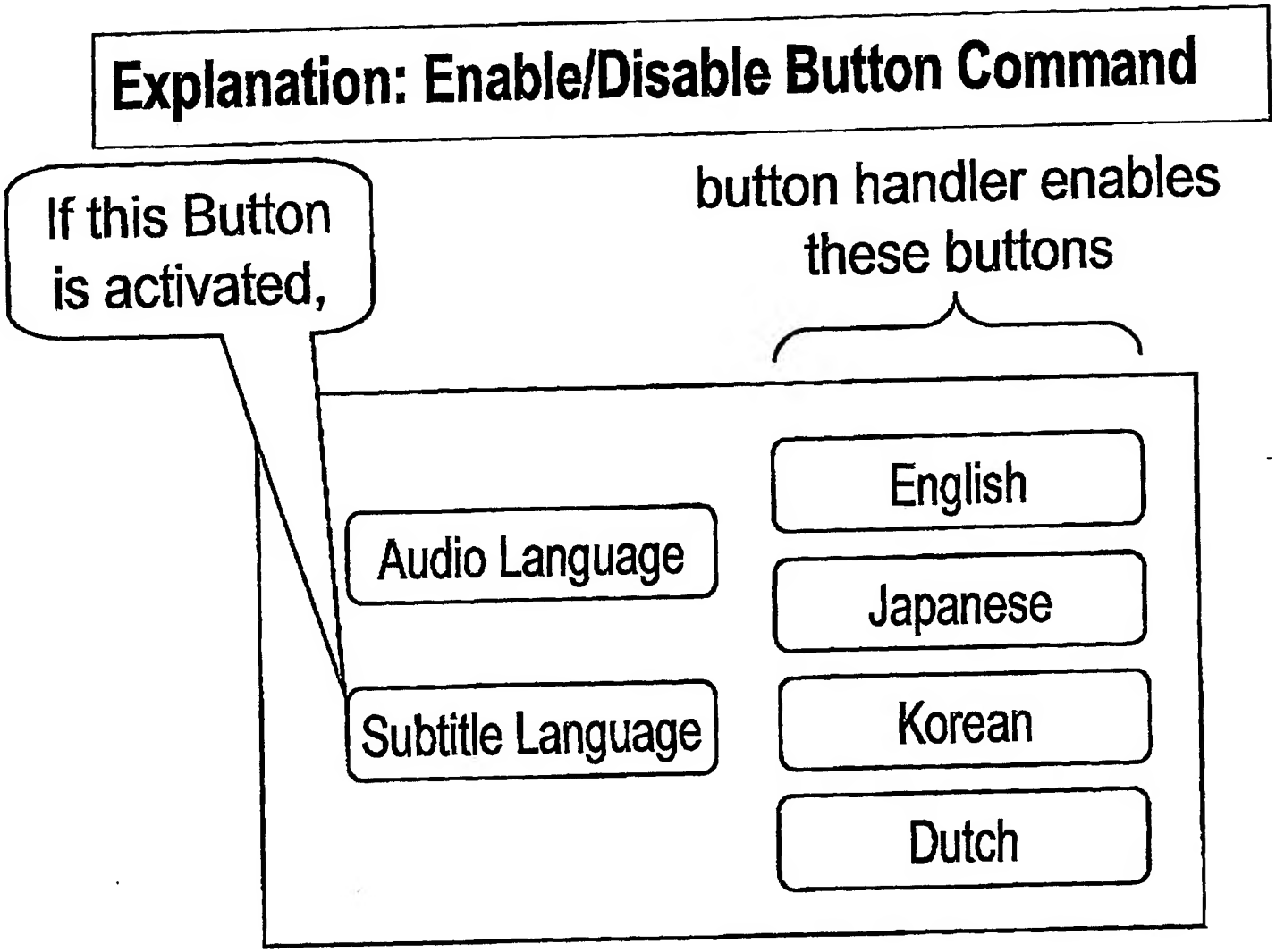


Neighborhood relations leading to disabled buttons are defined but the decoder ignores them as long as the button is disabled

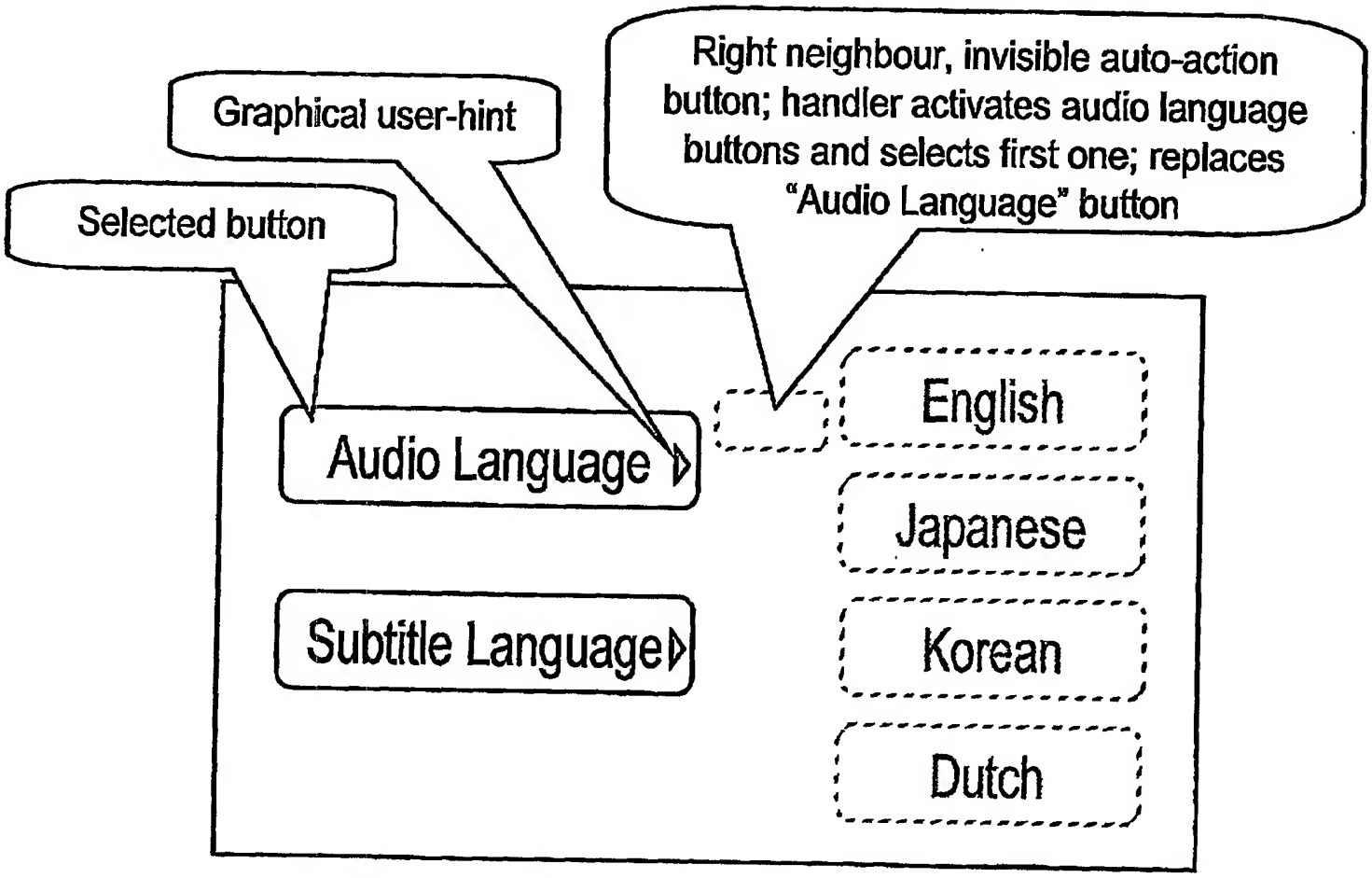
14. JAN. 2004 18:17 DTB PATENT DEPARTMENT GERMANY NR. 022 S. 24
EPO-BERLIN
14-01-2004

Explanation: Enable/Disable Button Command



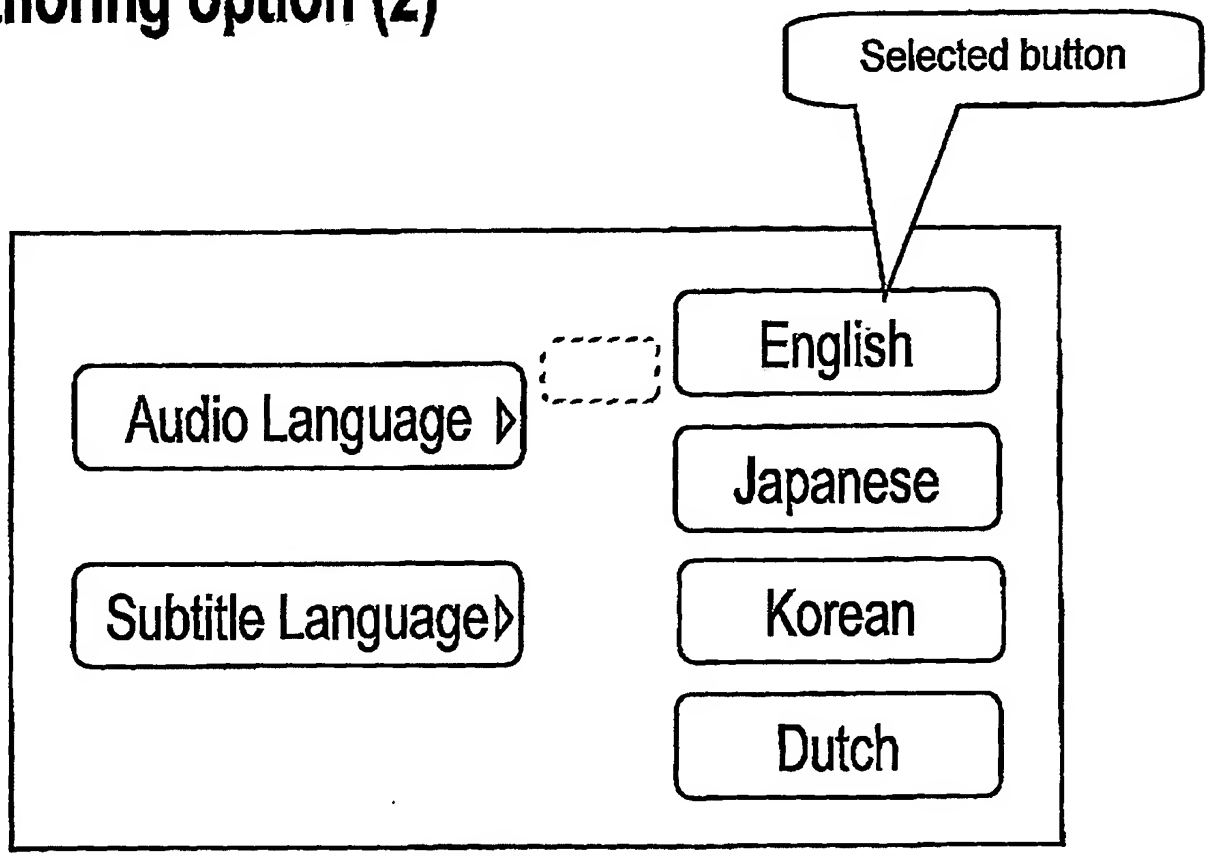

Explanation: Enable/Disable Button Command
If this Button is activated,
button handler enables these buttons
Audio Language
Subtitle Language
English
Japanese
Korean
Dutch

Authoring option (1)



14. JAN. 2004 18:17 DTB PATENT DEPARTMENT GERMANY NR. 022 S. 27

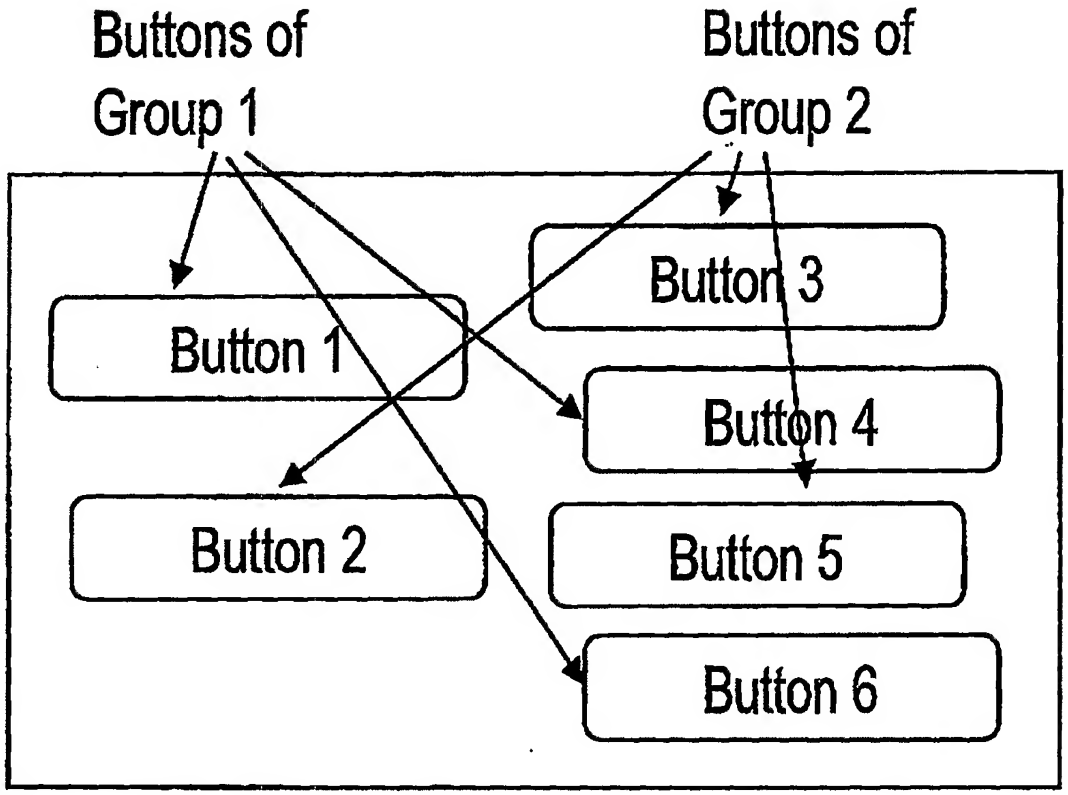
Authoring option (2)



14. JAN. 2004 18:17 DTB PATENT DEPARTMENT GERMANY NR. 022 S. 28

Button group area – 1st definition

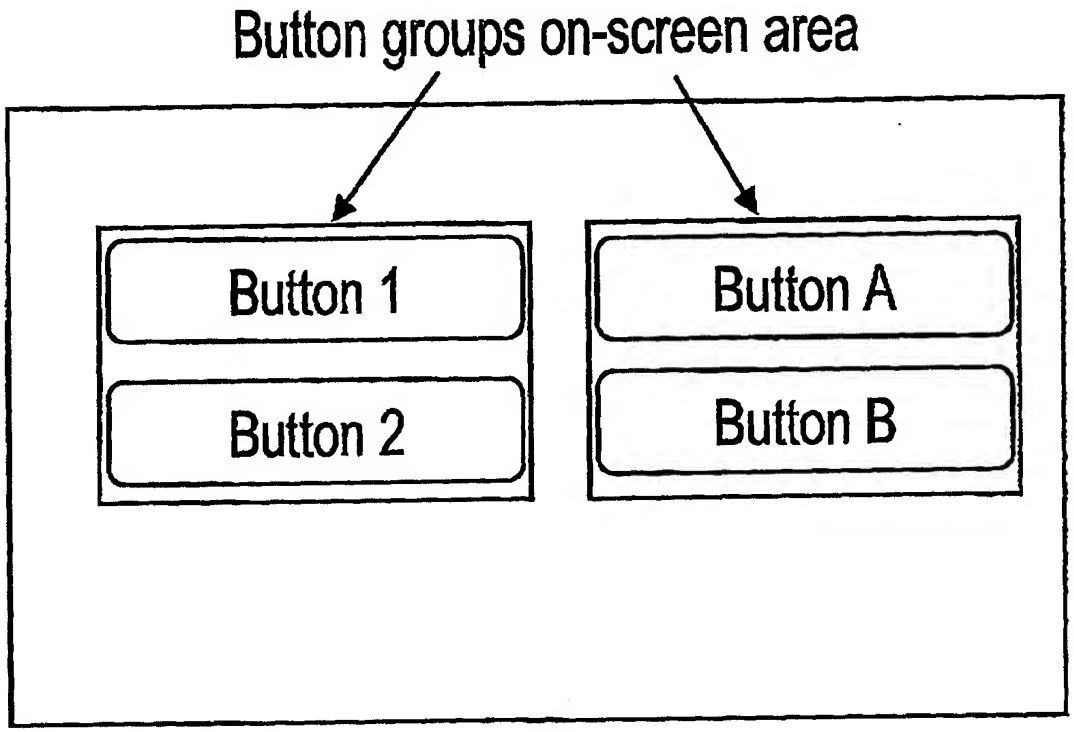
- 1st definition: The button group area is defined through the sum of all buttons' on-screen areas of the button group



14. JAN. 2004 18:17 DTB PATENT DEPARTMENT GERMANY NR. 022 S. 29

Button group area – 2nd definition

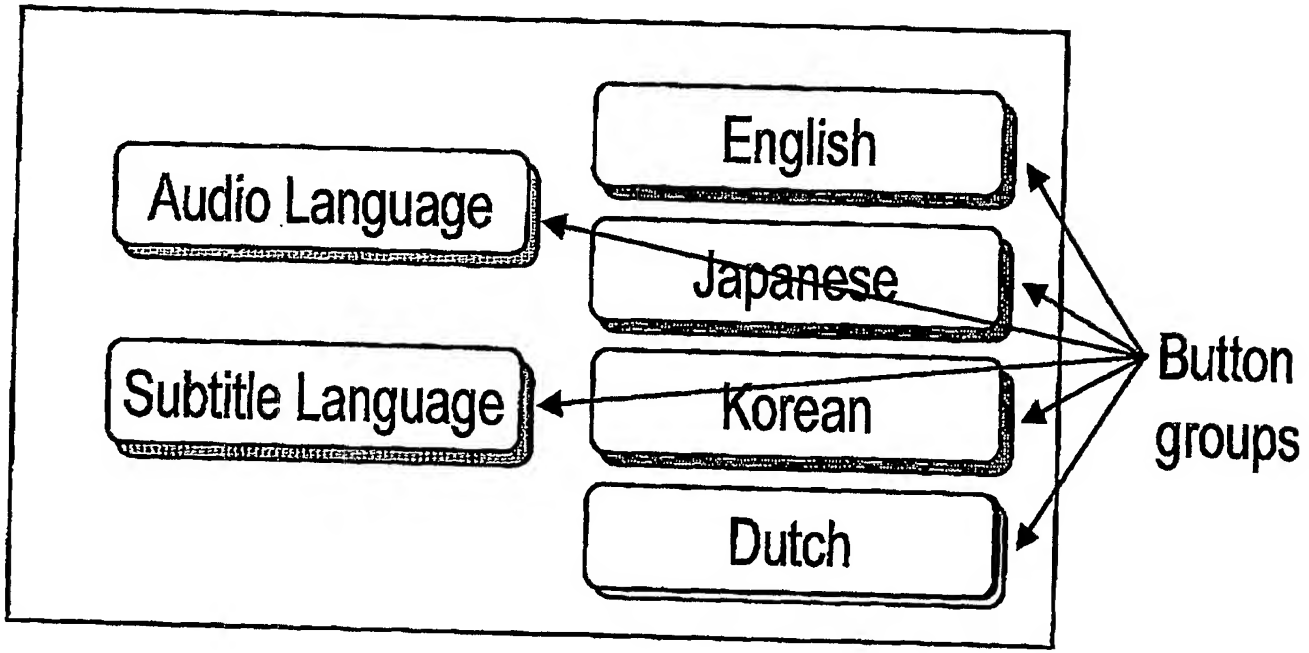
- 2nd definition : A group is assigned a rectangular on-screen area. All buttons' on-screen areas lie within the group's on-screen area



14. JAN. 2004 18:17 DTB PATENT DEPARTMENT GERMANY NR. 022 S. 30

Button groups area – 3rd definition

- 3rd definition : All buttons in a group occupy the same on-screen area which is at the same time the group area;



14. JAN. 2004 18:17 DTB PATENT DEPARTMENT GERMANY NR. 022 S. 31

Verification Aspects (1)

- Neighborhood relationships are statically defined
- Button positions are statically defined
- Buttons can overlap but only one of them is allowed to be enabled at any given time

14. JAN. 2004 18:17 DTB PATENT DEPARTMENT GERMANY NR. 022 S. 32

Verification Aspects (2)

- We define a Button Group as a group of buttons, where only one of the buttons in the group can be in "enabled" state
- Buttons within the same group are allowed to have overlapping positions on the screen
- Multiple Button Groups are allowed, but no button belonging to Button Group A may overlap with any button belonging to any of the other Button Groups
- The new "Enable/Disable" navigation command specifies the Button Button Number
  - Enable Button x
  - Disable Button x

14. JAN. 2004 18:17 DTB PATENT DEPARTMENT GERMANY NR. 022 S. 33